

Basic Electronics/Programming

Jordan Wade
2007 Oklahoma Robotics Summit

1

Electronics and Programming presentation given at the 2007 Oklahoma Robotics Summit.

Wiring

- Wire sizes
- Crimp connectors
- Soldering
- Insulation

2

Wire sizes: Use the correct size wire for the load. Don't use PWM size wire for a CIM motor.

Crimp connectors: Use the correct size connector for the wire. Don't use a connector made for 8 AWG wire on 14 AWG wire.

Soldering: Use a good quality soldering iron and take care of it. Check all solder joints after soldering. High temperatures for a short time is better than lower temperatures for a longer time.

Insulation: Insulate all wiring using either electrical tape or heat shrink.

Components

- Robot Controller
- Compressor
- Motors
- Relays
- Sensors

3

These are some of the components that you will have wiring going to.

Power Distribution

- Batteries
- Main breaker
- Power distribution blocks
- Circuit breakers

4

Refer to the Power Distribution Diagram put out by FIRST each year for the required power distribution wiring. It is made available at kick-off.

PWM Cables

- Sensor inputs
- Speed controllers
- Relays
- Servos
- Camera



PWM cables are used for almost all data.

FRC does *not* use keyed cables (i.e. it is possible to plug the cable in backwards). Check to make sure the cable is plugged in correctly before powering up.

Programming

- easyC
- WPILib
- IFI Default Code

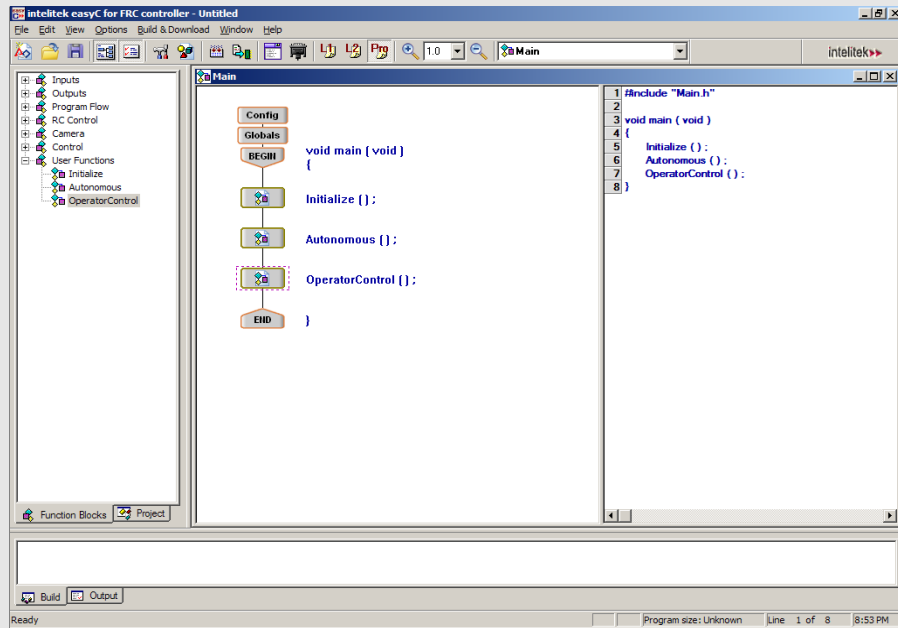
6

easyC: Recommended for rookie teams. Graphical drag-and-drop.

WPILib: Recommended for more experienced teams. Text-based, the backend for easyC.

IFI Default Code: Recommended for older teams only. Most difficult and lowest level of the options.

easyC



7

easyC after creating a competition project.

easyC Programming

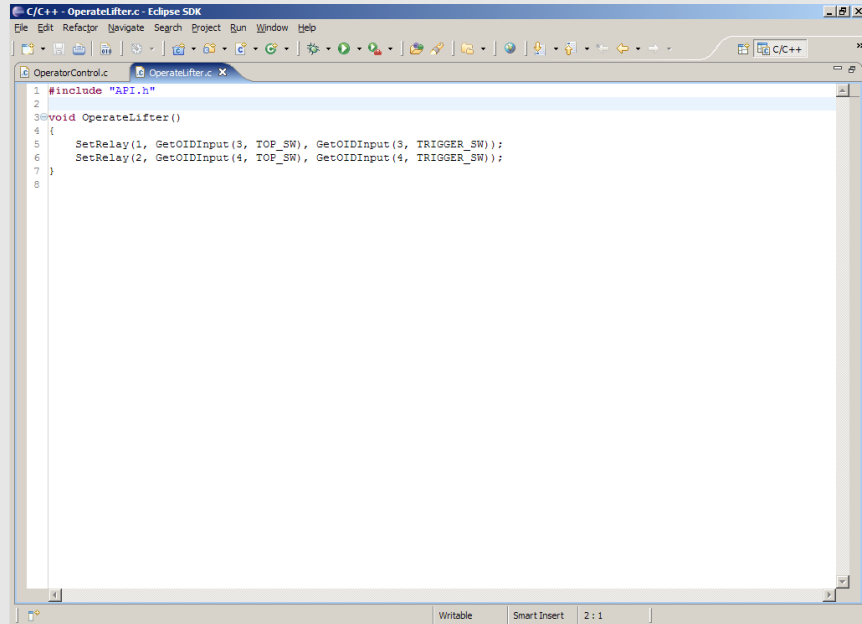
- Drag-and-drop
- Built-in functions for common tasks
 - Drive motors
 - Sensors
- Easy to get started
- Included in kit

8

easyC is the highest level of the options and the easiest to get started with.

One license is included in the kit of parts per team. To use easyC on more than one computer, you must buy additional licenses.

WPILib



The screenshot shows the Eclipse IDE interface with a C++ project open. The main editor window displays the following code:

```
1 #include "API.h"
2
3 void OperateLifter()
4 {
5     SetRelay(1, GetOIDInput(3, TOP_SW), GetOIDInput(3, TRIGGER_SW));
6     SetRelay(2, GetOIDInput(4, TOP_SW), GetOIDInput(4, TRIGGER_SW));
7 }
8
```

The IDE window title is "C/C++ - OperateLifter.c - Eclipse SDK". The status bar at the bottom indicates "Writable", "SmartInsert", and "2 : 1".

9

Eclipse with a WPILib project open. Code is from the lifter on 476's 2007 robot.

WPILib Programming

- Used in easyC
- Allows lower-level access
- Updated more often than easyC

10

WPILib provides lower level access and more features than easyC.

WPILib is updated more often than easyC, adding more features or fixing bugs.

IFI Default Code

```
C:\team476-2006\trunk\Loader.c - Notepad2
File Edit View Settings 2
// Loader.c
#include <stdio.h>
#include "Loader.h"
#include "ifi_aliases.h"
#include "ifi_default.h"

//unsigned char rom LoaderGateLoad [6] = {255,173,200,247,213,210};
//unsigned char rom LoaderGateRelease [6] = {162,73,96,143,116,119};
unsigned char rom LoaderGateLoad [6] = {LOADER_UPPER_LEFT_CAL_OFFSET + LOADER_CAL_RANGE,
    LOADER_UPPER_CENTER_CAL_OFFSET + LOADER_CAL_RANGE,
    LOADER_UPPER_RIGHT_CAL_OFFSET + LOADER_CAL_RANGE,
    LOADER_LOWER_LEFT_CAL_OFFSET + LOADER_CAL_RANGE,
    LOADER_LOWER_CENTER_CAL_OFFSET + LOADER_CAL_RANGE,
    LOADER_LOWER_RIGHT_CAL_OFFSET + LOADER_CAL_RANGE };

unsigned char rom LoaderGateRelease [6] = {LOADER_UPPER_LEFT_CAL_OFFSET,
    LOADER_UPPER_CENTER_CAL_OFFSET,
    LOADER_UPPER_RIGHT_CAL_OFFSET,
    LOADER_LOWER_LEFT_CAL_OFFSET,
    LOADER_LOWER_CENTER_CAL_OFFSET,
    LOADER_LOWER_RIGHT_CAL_OFFSET};

unsigned char LoaderTimer;
LoaderFlags_t LoaderFlags;

void _LoaderSetServoPosition(char gate, char load)
{
    unsigned char position;
    if (load) position = LoaderGateLoad[gate];
    else position = LoaderGateRelease[gate];
    printf("Servo position: %d\r", position);
    switch (gate)
    {
        case LOADER_UPPER+LOADER_LEFT:
            LOADER_UPPER_LEFT_CHANNEL = position;
            break;
        case LOADER_UPPER+LOADER_CENTER:
            . . .
    }
}
```

11

Notepad2 with a default code project open. Code is from the shooter 476's 2006 robot.

IFI Default Code Programming

- Would *not* recommend for new teams
- Lowest-level access



Default code provides the lowest level access for teams that need it.